Super Simple Hook

Ian Wasser, Robin Steiner

Neue Kantonsschule Aarau

Infcom

Inhaltsverzeichnis

[Konzept: 2](#_Toc62331976)

[Inspiration: 2](#_Toc62331977)

[Milestones: 3](#_Toc62331978)

[Basis-Version: 3](#_Toc62331979)

[Player Movement: 3](#_Toc62331980)

[Hook Mechanic: 3](#_Toc62331981)

[World Objects (+ Camera): 3](#_Toc62331982)

[Maps: 3](#_Toc62331983)

[Design: 4](#_Toc62331984)

[UI: 4](#_Toc62331985)

[Vorgehensweise: 5](#_Toc62331986)

[Reflexion: 6](#_Toc62331987)

[Ian Wasser: 6](#_Toc62331988)

[Robin Steiner: 7](#_Toc62331989)

[Weiteres nach der Abgabe: 8](#_Toc62331990)

[Attribution: 9](#_Toc62331991)

[Code Snippets: 9](#_Toc62331992)

[Musik: 9](#_Toc62331993)

Super Simple Hook

Created by: Robin Steiner, Ian Wasser

# Konzept:

Unser Spiel “Super Simple Hook” ist ein 2D Puzzle Plattformer Game. Das Ziel des Spiels ist es, durch das Lösen von Rätseln den Ausgang eines Raumes / Levels zu erreichen (wie ein Escape Room). Für das Lösen der Rätsel steht dem Spieler ein Grappling Hook zur Verfügung. Dieses Feature bildet das Hauptelement, welches unser Spiel von anderen Puzzlegames abhebt.

Mit dem Grappling Hook kann sich der Spieler, durch Drücken der Maustaste, an festgelegte Objekte (orange Flächen) anheften und dort hinziehen lassen. Zusätzlich kann der Spieler auch bewegbare Objekte, wie Würfel und Kugeln, an diese orangen Targets heranziehen lassen.

Das Spiel sollte vom Design/Art her, wie der Name sagt, möglichst simpel sein. Man sollte nicht durch irgendwelche Designelemente von den Puzzles abgelenkt werden.

# Inspiration:

Bei unserem Spiel liessen wir uns durch das 2007 von Valve veröffentlichte Spiel «Portal» inspirieren. «Portal» ist eines unserer Lieblingsspiele und unserer Meinung nach ein Must-Have für jeden puzzle-liebenden Spieler. In dem Spiel muss der Spieler Räume mit Rätsel lösen, mithilfe einer Portal-Gun.

Für unser Spiel übernahmen wir die Idee, von einzelnen Rätselräumen und einem langsam aufbauenden Tutorial, welches der Spieler gar nicht bemerkt. Ein grosser Unterschied zum Original ist, dass unser Spiel 2D ist.

Bei der Idee vom Grappling Hook (anstelle der Portal-Gun) liessen wir uns von der Spielereihe «Just Cause» inspirieren. Dort bildet der Hook ein Nebenelement, ist aber trotzdem ein sehr witziges und zeitvertreibendes Feature.

Unsere Idee besteht somit hauptsächlich aus bereits bestehenden Elementen, die in Kombination aber ein neues Spiel erschaffen.

# Milestones:

## Basis-Version:

Bis: 16.12.2020

Die Basisversion soll aus den Player/Hook-Mechaniken bestehen. Ausserdem sollten einzelne Objekte, wie die Targets für den Hook und die Objekte, die die Welt bilden, hinzugefügt werden. So wäre das Spiel bereits spielbar. Ausserdem soll die 2D-Physics-Engine Matter.js hinzugefügt werden.

### Player Movement:

Bis: 13.12.2020

Die Physik Engine Matter.js soll dem Spiel hinzugefügt werden und einfache Bedienungsmöglichkeiten, wie das Springen und Laufen auf dem Boden ermöglichen.

### Hook Mechanic:

Bis: 13.12.2020

Der Hook sollte dort hingeworfen werden, wo die Maus hinzeigt. Sobald der Hook ein Target berührt, soll der Hook nicht mehr geworfen werden und stattdessen den Player anziehen. Wenn der Hook ein anderes Objekt berührt, soll er gelöscht werden.

Update: 25.12.2020

Neues Hook-Feature: Wenn die Maus gedrückt gehalten wird, wird der Player nicht angezogen. Stattdessen bleibt der Hook am selben Ort und man kann einen zweiten Hook abfeuern. So kann man andere Objekte als den Player anziehen lassen.

### World Objects (+ Camera):

Bis: 13.12.2020

Zuerst sollen nur Targets und normale Objekte hinzugefügt werden. So ist das Spiel bereits spielbar.

Update: 25.12.2020

Button, Tür, und bewegbare Objekte sollen hinzugefügt werden, damit Rätsel möglich sind.

Update: 30.12.2020

Loadtriggers, die das nächste Level laden und Deathtrigger, die bei Berührung das Level neustarten, sollen hinzugefügt werden.

## Maps:

Bis: 27.12.2020

Die Level sollen in einem Array in einer JSON Datei gespeichert werden. Durch ein Manager-Script sollen dann diese Arrays ausgelesen werden und das Level, das darin steckt, generiert werden.

Die JSON Dateien sollen durch einen eigenen Levelditor erstellt werden. Darin kann man Objekte (und später auch Tiles) platzieren. Sobald ein Objekt platziert wird, wird es auch direkt im Array gespeichert. Der Leveleditor ist nur ein Developer-Tool und wird nicht mitabgegeben.

Trotzdem hat der Leveleditor sehr viel Aufwand erspart und ohne ihn wäre das Spiel in dieser Form nicht möglich gewesen.

## Design:

Bis: 3.1.2021

Einzelne Tiles sollen mit Adobe Illustrator designt werden. Diese sollen nach Möglichkeit mithilfe des Leveleditors selbst platziert werden können. Das ermöglicht ein freies und individuelles Leveldesign.

Update: 16.1.2021

Grafiken sollen für alle Objekte hinzugefügt werden. Der Player soll animiert werden. Wir haben uns dazu entschieden, keine Beine zu machen und das Movement stattdessen anders zu simulieren, weil das Spiel möglichst simpel bleiben soll.

## UI:

Bis: 10.1.2021

Ein Pause-Menu soll hinzugefügt werden. Hauptzweck davon soll sein, das Level neu zu starten. Man soll auch zum Main Menu zurückkehren können.

Main Menu: Man soll ein neues Spiel starten können, aber man soll auch das Spiel weiterspielen können. Für diesen Zweck wird der Fortschritt automatisch gespeichert.

# Vorgehensweise:

Zuerst machten wir uns einige Gedanken, wie das Spiel allgemein aussehen sollte. Dies brauchte nicht viel Zeit und wir wurden uns schnell einig. Wir wollten ein Spiel, das interessant als auch fordernd für den Spieler war.

Danach richteten wir das Source Control Programm „Git“ mit dem Server „GitHub“ ein. Das Programm half uns bei einigen kleinen Problemen, die auftraten.

Danach waren wir bereit mit dem Programmieren zu beginnen. Wir richteten uns eine Lokale Umgebung ein, wo wir einfach programmieren konnten, da „Openprocessing“ unserer Meinung nicht die beste Umgebung zum Programmieren ist.

Zuerst arbeiteten wir etwas unorganisiert darauf los. Jeder machte einen Teil, auf den er Lust hatte. Dank „Git“ konnten wir an einzelnen Features arbeiten, ohne auf Probleme zu stossen. So entstanden langsam die ersten Grund-Features.

Nachdem die Grund-Features fertiggestellt wurden, kamen wir an den wahrscheinlich schwierigsten Punkt in der Entwicklung. Wir wollten einen Level-Editor, womit wir schnell und einfach neue Level gestalten konnten. Diese Aufgabe stellte sich etwas schwerer heraus, als wir dachten. Dennoch schafften wir es nach einiger Zeit des programmierens und einer genauen Besprechung der Vorgehensweise, den Leveleditor fertigzustellen. Mit ihm hatten wir es später leichter die einzelnen Level zu gestallten.

Danach kümmerten wir uns um die Grafiken, welche wir mithilfe von Illustrator gemacht haben. Da wir beide nicht die grössten Designer sind und auch unser Ziel ein möglichst simples Spiel sein sollte, machten wir einfache Vektorgrafiken. Da wir aber nicht einfach nur einheitliche Bilder haben wollten, kam der schwierige Teil, nämlich Tilesets einzuführen.

Mit der Zeit kamen immer mehr Designs für die verschiedensten Objekte dazu. Nun waren wir aber auch an dem Punkt angekommen, wo wir ein Interface brauchten. Somit fügten wir ein recht allgemeines Interface ein, um uns etwas Code zu sparen. Auch das musste designt werden.

Nachher fügten wir noch Musik ins Spiel ein, welche wir kostenlos von „Tim Beek“ verwendeten. Die Musik war lizenzfrei und passte recht gut zu unserem Spiel.

Um dann das grundsätzliche Spiel Abzuschließen, brauchten wir noch Tutorials, um dem Spieler den Weg zu zeigen. Dies wollten wir mithilfe vom Leveldesign machen, dass die Rätsel stark aufeinander aufbauend sind. Da das Spiel aber so leicht verständlich wie möglich sein sollte, entwickelten wir kleine Schilder, die dem Spieler die Steuerung beibrachten.

Nach dem all diese Dinge fertiggestellt waren, wollten wir noch eine Reihe von Beta Tests machen. Wir schickten eine Umfrage und das Spiel an viele Leute, die wir kannten. Grundsätzlich waren die Tester begeistert von dem Spiel und wollten gerne noch mehr davon sehen. Das grösste Problem, was die Tester hatten, war, dass die Steuerung im Spiel nicht ganz einfach zu verstehen war. Darum entschieden wir uns, die Schilder noch zu animieren, um noch etwas mehr Information einzubauen.

Ein weiters, grosses Problem war die Performance. Wir versuchten lange den Grund für die Probleme zu finden, aber ohne Erfolg. Darum begannen wir es so gut es geht zu verhindern. Der beste Fix war zum Schluss der, dass wir beim Levelwechsel die Seite neu laden. So stabilisierte sich die Performance bei jedem Levelwechsel.

Danach entschieden wir uns, dass das Spiel nun für die Abgabe bereit war.

# Reflexion:

## Ian Wasser:

Grundsätzlich denke ich, haben wir das Spiel vor allem in Hinsicht auf die steigende Schwierigkeit recht gut hinbekommen. Die Level sind nicht zu schwer, als auch nicht zu einfach. Sie sind sogar laut Testpersonen fast etwas zu einfach. Dadurch wird es möglich, dass auch nicht geübte Gamer das Spiel trotzdem spielen können. Auch das Design ist uns gut gelungen. Es passt super zum Spiel und ist sehr simpel.

Grundsätzlich ist auch die Programmierung schnell voran gegangen und wir konnten schnell Erfolge erzielen. Vor allem die Grundfeatures waren schnell fertig. Leider haben wir uns etwas stark bei dem Aufwand vom Leveleditor und vom Polishing verschätzt. Der Leveleditor war dann doch etwas schwieriger als gedacht und brauchte viel Zeit. Dennoch hat sich der Aufwand gelohnt, da unser Spiel in dieser Grössenordnung nicht möglich gewesen wäre.

Da ich noch nie einen Leveleditor gemacht hatte, überlegte ich mir wie ich das ganze programmieren könnte. Ich nahm aber keine Rücksicht darauf, wie das jetzige Map System von Robin funktionierte. Robin hatte alle Objekte in Gruppen in die verschiedenen Arrays aufgeteilt. Ich wollte ein einziges Array mit allen Objekten haben. Ich hatte zum Schluss das Problem, dass ich ein spezifisches Objekt aus der Welt im Array finden musste und verändern wollte. Ich kam recht lange nicht auf die Idee meine Ansicht auf die von Robin umzustellen, was mich viel Zeit kostete. Nun weiss ich es besser. Ich sollte nicht nur mein System verfolgen, sondern mein System auch auf andere anpasse. Dies war eine wichtige Lektion für mich, da ich zuvor noch nie richtig in dieser Grössenordnung mit jemanden zusammengearbeitet habe.

Beim Polisching kostete uns vor allem das Angleichen auf alle Screens einiges an Zeit. Wir versuchten, das Spiel auf allen Computern (egal welcher Auflösung oder Verhältnis) gleich aussehen und laufen zu lassen. Ein Problem war sicher die Physikberechnungen, die bei grösseren Monitoren anders verlief als bei kleinen. Mit vielen kleinen Zahlenumrechnungen konnten wir diese Probleme aber auch lösen.

Dabei verschätzten wir uns etwas mit dem Aufwand alles zu säubern und zu verfeinern. Wir schafften es trotzdem, weil wir uns viel Puffer bis zur Abgabe gesetzt hatten, aber für weitere Kleinigkeiten, die wir gerne noch eingefügt hätte, hat es dann nicht mehr gereicht. Diese Sachen werden wir dann auf die Zeit nach der Abgabe verschieben.

Ein sehr wichtiger Punkt, den ich unbedingt das nächste Mal besser machen möchte, ist die Organisation. Wir haben das Projekt zuerst unorganisiert begonnen und machten beide zwei verschiedene Sachen. Irgendwann ist uns klar geworden, dass es nicht ganz so gut weiter funktioniert und wir gingen unbewusst nach den Scrum System vor. Wir trafen und ein oder 2-mal die Woche und besprachen, was jeder bis zum nächsten Mal macht. Grundsätzlich hat dieses System sehr gut funktioniert, dennoch würde ich das nächste Mal versuchen noch ein Todo-Listen Programm miteinzubeziehen, damit wir dort unsere Aufgaben besser verwalten können.

Auch wäre eine genauere Planung am Anfang sehr wichtig gewesen, was wir genau wie im Spiel haben wollen.

Grundsätzlich bin recht zufrieden mit unserer Leistung. Unsere Testpersonen, die zum Teil auch schon andere Spiele von mir getestet haben, waren sehr begeistert und sagten, dass dies das beste Spiel ist, welches ich je gemacht habe. Darum werde ich mit Robin noch weiter an dem Spiel und auch noch an anderen Spielen arbeiten. Auch denke ich, konnten wir voneinander recht gut profitieren.

## Robin Steiner:

Im Gegensatz zu Ian, war dieses Spiel für mich das erste grössere Programmierprojekt, und ich habe deswegen viele neue Erfahrungen gemacht. Vor allem im Bezug auf Organisation ist mir einiges klar geworden. Wir haben zwar zusammen ein Konzept herausgearbeitet und schon im Voraus einige Milestones gesetzt, aber als es dann ums Programmieren ging, machte jeder ein wenig das, worauf er gerade Lust hatte. Das klingt zwar nicht so, als wäre es eine gute Herangehensweise, aber weil wir uns immer wieder online getroffen haben und unsere Fortschritte besprachen, hat es schlussendlich doch funktioniert. Mir ist die also die Wichtigkeit der Kommunikation klar geworden. Ausserdem habe ich den Umgang mit Git gelernt, was mir vielleicht noch viel nützen wird.

Mit dem Produkt bin ich grundsätzlich sehr zufrieden. Auch wenn es an einigen Stellen einige Makel aufweist, sieht es ungefähr so aus, wie ich es mir am Anfang vorgestellt habe. Auch mit dem Konzept bin ich zufrieden. Trotz schon bekannten Elementen existierte das Spiel in dieser Form noch nicht. Und bei der riesigen Menge an Spielen, die auf dem Markt sind, ist das schwer zu erreichen.

Ich war während der ganzen Produktion des Spiels hoch motiviert und habe so gut wie jeden Tag etwas programmiert, designed und so weiter. Beim Programmieren haben wir schnell Fortschritte gemacht aber das Designen fiel mir schwerer. Ich kannte mich mit Illustrator recht gut aus aber hatte häufig einen Mangel an Ideen. Normalerweise lief bei mir das Designen so ab, dass ich eine Zeit lang versuchte, etwas Beeindruckendes zu kreieren, bis ich mich irgendwann dazu entschied, etwas Simpels zu machen. Etwas, was ich auch wirklich hinkriege.

Es gab manchmal auch Zeiten, wo ich nicht mehr genau wusste, was ich überhaupt machen sollte. Das war meistens der Fall, wenn ich gerade irgendein Feature/Element fertiggestellt habe. So etwas wie eine genauere Todo-Liste, als die, die wir hatten, wäre hier hilfreich gewesen.

Ich nehme mir für die Zukunft, nach den eher schlechten Erfahrungen in diesem Projekt, eine bessere Organisation vor. Wie schon gesagt hat hier schlussendlich doch alles recht gut funktioniert. Ich glaube aber, dass ich durch bessere Organisation mindestens meine eigene Effizienz deutlich steigern könnte. Einer der Hauptgründe, warum das Spiel so herauskam, wie ich es mir vorgestellt hatte, war mein Teampartner. Ich glaube wir konnten beide voneinander profitieren. Unter anderem konnte ich von seiner Programmiererfahrung profitieren.

Ich selbst habe erst in Kantonsschule das Programmieren gelernt und habe in diesem Semester mit jedem einzelnen Projekt dazugelernt. So konnte ich nun in diesem letzten Projekt mit ihm darüber diskutieren, wie wir das Spiel programmieren. Ausserdem konnte ich mir seinen Code ansehen und versuchte ihn nachzuvollziehen. Ich konnte auch nach möglichen Verbesserungen suchen. So habe ich viele Tricks im Programmieren gelernt.

Für die Abgabe, finde ich, macht es keinen grossen Sinn, das Spiel weiter auszubauen. Aber wir haben beide Interesse daran, das Spiel nach der Abgabe weiterzuentwickeln. Besonders eher kleine Dinge wie die Performance-Probleme (Die nur durch eine Notlösung reduziert wurden) sind mir noch immer ein Dorn im Auge. Wenn das Spiel irgendwann mal in einem Zustand ist, wo es tatsächlich mit professionelleren Indie-Games vergleichbar ist, könnten wir es vielleicht sogar veröffentlichen.

# Weiteres nach der Abgabe:

Da unsere Testpersonen sehr begeistert von dem Spiel waren und sich noch mehr von dem Spiel wünschen, wollen wir nach der Abgabe privat das Spiel noch etwas weiterführen. Ein Ziel von uns wäre es noch einige Bugs zu fixen, weitere Level noch hinzuzufügen und auch noch weitere Kleinigkeiten zu verbessern.

Das Ziel wäre dann das Spiel als fertige Client Applikation auf dem Computer auf diversen Plattformen zu veröffentlichen. Da wäre die Möglichkeit Steam oder der Microsoft Store.

# Attribution:

## Code Snippets:

<https://stackoverflow.com/questions/7346563/loading-local-json-file>

<https://www.w3schools.com/js/js_json_http.asp>

<https://stackoverflow.com/questions/32283510/closest-value-snapping>

<https://stackoverflow.com/questions/34156282/how-do-i-save-json-to-local-text-file>

<https://www.mediaevent.de/javascript/string-search.html>

<https://stackoverflow.com/questions/1621891/html-how-to-make-a-submit-button-with-text-image-in-it>

<https://stackoverflow.com/questions/22672368/how-to-make-a-transparent-html-button>

Die oben aufgeführten Links wurden für Hilfezwecke, Inspiration von Algorithmen oder ähnlichem verwendet.

## Musik:

Musik created by Tim Beek: <https://timbeek.com/>